Name: Samuel Foley
Date: February 14, 2024
Course: IT FDN 110A
Github: https://github.com/Samuel-a-m-f/IntroToProg-Python-Mod05

Assignment 05 – Advanced Collections and Error Handling

Introduction

This document is going to describe the steps that I took for this week's assignment. This assignment builds on Assignment 4 and deals with JSON files instead of CSV files, the use of dictionary as a place to store data, and exception handling. This will not cover the code previously discussed on Assignment 4, simply the changes that were made for assignment 5.

Topic

The first section in my script covers the data constants and will use only strings. The two constants that are defined along with the header are going to be the FILE_NAME and MENU. The important aspect here was the use of import json to be able to read, write and manipulate JSON files. The file that will be read is a json file called Enrollments.

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   S. Foley,2/12/2024, Assignment 5 Script Creation
# ------------------------------------------------------------------------------------------ #

import json

# Define the Data Constants
MENU: str = '''
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------
'''
# Define the Data Constants
FILE_NAME: str = "Enrollments.json"
```

Topic

The next section of code defines the variables. The variables all stayed the same except for student_data which was changed to a dict and uses {} to define a dictionary instead of the list that was used before.

```
# Define the Data Variables and constants
student_first_name: str = ''  # Holds the first name of a student entered by the user.
student_last_name: str = ''  # Holds the last name of a student entered by the user.
course_name: str = ''  # Holds the name of a course entered by the user.
student_data: dict = {}  # one row of student data
students: list = []  # a table of student data
csv_data: str = ''  # Holds combined string data separated by a comma.
file = None  # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user.
```

The json file was read using the same open as before, but uses json.load to dump the data into students. This is also one of the first error handling event. An attempt will be made to open the file, if it does not find a file, it will spit out the first line “Text file must exist before running this script” and iterates to a non-specific error, or if the file does not close properly it will try again.

```
# When the program starts, read the file data into a list of lists (table)
# Extract the data from the file
try:
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()

except FileNotFoundError as e:
    print("Text file must exist before running this script!\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
except Exception as e:
    print("There was a non-specific error!\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
finally:
    if file.closed == False:
        file.close()
```

Topic

Similar to Assignments 3 and 4, this script uses a while loop which continues until the user breaks out by entering 4. The while loop is set to true and will continue if 1, 2, 3, or anything else that is not a 4.

```python
# Present and Process the data
while (True):

    # Present the menu of choices
    print(MENU)
    menu_choice = input("What would you like to do: ")

    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!
        student_first_name = input("Enter the student's first name: ")
        if not student_first_name.isalpha():
            raise ValueError("The first name should not contain numbers.")
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError("The last name should not contain numbers.")
        course_name = input("Please enter the name of the course: ")
        student_data = {"FirstName": student_first_name, "LastName": student_last_name, "CourseName": course_name}
        students.append(student_data)
        print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
        continue

    # Present the current data
    elif menu_choice == "2":

        # Process the data to create and display a custom message
        print("-"*50)
        for student in students:
            print(f"Student {student['FirstName']} {student['LastName']} is enrolled in {student['CourseName']}")
        print("-"*50)
        continue
```

```
    # Save the data to a file
    elif menu_choice == "3":
        try:
            file = open(FILE_NAME, "w")
            json.dump(students, file)
            file.close()
            continue
        except TypeError as e:
            print("Please check that the data is a valid JSON format\n")
            print("-- Technical Error Message -- ")
            print(e, e.__doc__, type(e), sep='\n')
        except Exception as e:
            print("-- Technical Error Message -- ")
            print("Built-In Python error info: ")
            print(e, e.__doc__, type(e), sep='\n')
        finally:
            if file.closed == False:
                file.close()
        continue

    # Stop the loop
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, or 3")

print("Program Ended")

```

Topic

The primary difference between this script and Assignment 4 is the error handling for Option 1 and the use of the dictionary to store the data.

```
    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!
        student_first_name = input("Enter the student's first name: ")
        if not student_first_name.isalpha():
            raise ValueError("The first name should not contain numbers.")
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError("The last name should not contain numbers.")
        course_name = input("Please enter the name of the course: ")
        student_data = {"FirstName": student_first_name, "LastName": student_last_name, "CourseName": course_name}
        students.append(student_data)
        print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
        continue
```

There is an error handling event after the input of the first name and the last name to not contain numbers. If any numbers are present it will flag an error and the error message will appear. And the student data is stored using instead of 0, 1, 2, etc. the actual names of the variables which are “FirstName”, “LastName”, and “CourseName” in {} instead of []. The dictionary line created in student_data is appended to students.

Topic

If the user chooses option 2, it will display the students data that has been read from the JSON file as well as anything that has been added using option 1.

```
    # Present the current data
    elif menu_choice == "2":

        # Process the data to create and display a custom message
        print("-"*50)
        for student in students:
            print(f"Student {student['FirstName']} {student['LastName']} is enrolled in {student['CourseName']}")
        print("-"*50)
        continue
```

Topic

If the user chooses option 3, it will open a json file called “Enrollments.json” to write to, it will dump students data, close the file and display the data that was written. There is also error handling that is introduced here to check for invalid JSON formatting, additional error messages, as well as if the JSON file is not closed properly for some reason.

```
    # Save the data to a file
    elif menu_choice == "3":
        try:
            file = open(FILE_NAME, "w")
            json.dump(students, file)
            file.close()
            continue
        except TypeError as e:
            print("Please check that the data is a valid JSON format\n")
            print("-- Technical Error Message -- ")
            print(e, e.__doc__, type(e), sep='\n')
        except Exception as e:
            print("-- Technical Error Message -- ")
            print("Built-In Python error info: ")
            print(e, e.__doc__, type(e), sep='\n')
        finally:
            if file.closed == False:
                file.close()
        continue
```

Topic

If the user chooses option 4, it will terminate the while loop by using break.

```
    # Stop the loop
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, or 3")

print("Program Ended")

```

Topic

This topic covers the testing of the script to verify that all of the requirements that were laid out were satisfied. The testing will be done using PyCharm and showing the output of both the PyCharm and the json file as shown below:

```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 2
--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
--------------------------------------------------

---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 1
Enter the student's first name: Sam
Enter the student's last name: Foley
Please enter the name of the course: Python 100
You have registered Sam Foley for Python 100.
```

```
What would you like to do: 2
-------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Sam Foley is enrolled in Python 100
-------------------------------------------------

---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 3
```

```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 2
-------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Sam Foley is enrolled in Python 100
-------------------------------------------------

---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 4
Program Ended
```

The script was tested in CMD and worked identically to when it was run in Pycharm.

Summary

This document described the steps I took to write my first script in PyCharm for Assignment 04.